8. Olivia

Program Name: Olivia.java **Input File: olivia.dat**

Olivia is getting bored memorizing multiplication tables. The tables Olivia must memorize have N rows and M columns. The top-most row is row 1, and the left-most column is column 1. At the intersection of the R-th row and the C-th column is the value R * C.

To make working with the table more fun, Olivia likes starting at some cell in the table, and only moving down and right, exploring the numbers as she goes. Olivia's favorite number is K, and she enjoys it if she can get to a cell with value K from the starting cell.

After watching Olivia play for a little while, her parents were wondering how many different cells there are on an N by M multiplication table where you can walk to some cell with value K by only moving right and down. Since these tables can get massive, they've asked you to write a program to help them out.

Input: The first line of input is an integer T (1 ≤ T ≤ 50), the number of test cases. Each test case contains three space-separated integers N, M, and K. (1 ≤ N, M, K ≤ 1,000,000,000). N is the number of rows in the multiplication table, M is the number of columns in the multiplication table, and K is the target number.

Output: For each test case, output a single integer: the number of cells where there is at least one path from that cell to a cell with value K.

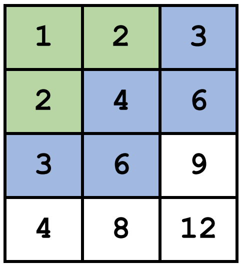
Sample input:

```
4
4 3 2
4 3 6
10 10 25
3141592 6535897 9323846
```

Sample output:

```
Case #1: 3
Case #2: 8
Case #3: 25
Case #4: 71686489
```

Sample explanation:
For the input `4 3 2`, the table is:



1	2	3
2	4	6
3	6	9
4	8	12

The “fun” cells are the green 1 and 2 in the top row, and the green 2 in the second row. For the input `4 3 6`, the table is the same as above, but any green or blue cell is “fun”. These are the first two rows and the first two cells in the third row.

In the third case, any cell in the first five rows and the first five columns is a valid starting point.